Exascale challenges and issues of applied mathematics

D.S.Butyugin[1,2], V.P.Il'in[1,2]

[1)] *The Institute of Computational Mathematics and Mathematical Geophysics, SB RAS,*
[2)] *Novosibirsk State Universuty, Novosibirsk, Russia,*
e-mail: ilin@sscc.ru, dm.butyugin@gmail.com

The current situation in HPC evolution gives computing community new possibilities to solve challenging direct and inverse interdisciplinary real-life problems, and new ground-breaking knowledges and technologies can be obtained through innovations in the mathematization of the researches in all areas of human activity.

To describe the corresponding approaches in the short thesis, we would present four incarnations of mathematics which play crucial role in this historical process.

At first, we should mention the recent remarkable results in the *theoretical mathematics*: topological methods, homological techniques, exterior algebra, shape derivatives and analysis, differential forms, Hamiltonian formalism, applied topics in group theory and functional analysis, see [1]–[5] for example, — which are still waiting own applications in many practical areas (electromagnetism, elasticity and structure analysis, hydrodynamics, chaos and catastrophes, etc.).

Secondly, these approaches initiate and give foundations for the new wave of advanced methods in *computational mathematics*, or *numerical analysis*: discrete forms, finite element exterior calculus [2, 6], symbolic computing, graph analysis, geometric numerical integration [7], simulating Hamiltonian dynamics [8], symplectic conservative algorithms, optimal control and inverse problems via optimization techniques, and with "classical" achievements of XX-th century, the advanced classes of algorithms are obtained for the main types of mathematical physics problems.

Of course, the modern numerical methods can not be considered today without scalable parallelization and mapping of algorithm structures on architectures of supercomputers with many millions of processors and cores, complex memory hierarchies, and communication networks. Algebraic and geometric domain decomposition, graph techniques, various communication avoiding tricks, and code optimization are the topics of many experimental researches and publications in the last decades, see discussions of some general and particular questions in [9].

The end unit in the discussed incarnation chain is *mathematical modeling*, or simulation, of processes and phenomena. Large scale computer experiments are accomplished mainly by means of complicated applied program packages (APPs), which are developed usually by big groups during several decades. There are many well-known commercial products (MSC NASTRAN, ANSYS, COMSOL, etc.) which can successfully handle multi-physics simulation in many applications. But the essential feature of these implementations is that they do not present the open innovations [10] and they are not oriented to wide extension and adaptation to particular situations by different users. Also, there are a lot of Open Source numerical tools which provide more or less suitability for the simulation problems, for example, FEniCS (finite element analysis), NETGEN (grid generation), METIS (graph analysis and partitioning), as well as numerous libraries for particular classes of algorithms. Roughly speaking, there is the great software zoo, and end user needs not only guides, but cumbersome manuals in order to solve own problems efficiently. The most important, end user needs too much time and high qualification in order to be able to use such low-level instruments, and it is practically impossible for the expert in the particular subject profession. Because of this reason the productivity of programmers and HPC simulations availability are bottlenecks as opposed to the fast grow of hardware performance.

In order to improve situation cardinally, we propose to input into considered incarnation chain the third link (intermediate, before the end one) – *computational integrated environment* (CIE) which would present the accessible set of numerical tools and technologies for supporting all main stages of mathematical simulation. In general, this activity includes the following steps [11]: geometric and functional modeling (interactive description of mathematical statement of direct and inverse problems, controlling the computational experiment); generating adaptive high quality grids, supporting local refinement and multigrid approaches; approximation of original problem to be solved using hierarchical high order constructions, finite element methods for example; solving the algebraic equations and eigenvalue problems using direct and preconditioned iterative algorithms for sparse matrices; optimal control and constrained local and global minimization; post-processing and visualization of the results; assembling configurations for particular applications; supporting the decision making.

In a sense, the set of these instrumental components presents the ecosystem which is not oriented to particular problem. But concrete APP can be gathered easy from such tools, similar to intellectual children constructor LEGO. It is important, that separate blocks of this environment can be developed independently by different groups, and internal mutual interfaces are provided by design of corresponding data structures, in accordance with N.Wirth's paradigm "Programs=Algorithms+Data Structure" [12].

Let us remark, that now there are some Internet publications on the prototypes of similar environment, Open FOAM [13] and Dune [14] for example. But these projects deal with considerable more restricted statements.

CIE should be oriented for a long life cycle, and it's architecture must satisfy several natural requirements:

- providing high resolution of numerical experiments; it means adaption and automatic guaranteed error control as well as the absence of program constraints on the number of degrees of freedom;
- mapping of algorithm structures on heterogeneous computer platforms and transparent scalable parallelization on all steps of computtational experiments, with no program constraints on the number of nodes, processors and cores; in a way, this characteristic is equivalent to adaptation of scientific software to continuous evolution of hardware;
- extendability of the set of mathematical models, algorithms and technological tools, a very important feature, because the development of the new mathematical statements and numerical methods is continuous and very dynamic process;
- compatibility with external applications and libraries via data structures conversions providing means to reuse existing software tools as building blocks of CIE as well as to export CIE parts into other projects;
- support of domain specific language and metamodel programming concepts [15]; this can substantialy improve product usability and promote broad acceptance within community.

These requirements are only basic ones, and many additional questions will arise later, during evolution of the concept and business plan of exascale technologies for broadening mathematical modeling in industry and knowledge mining.

Let us emphasize that such CIE cannot be handled by a single group, but only by scientific and industry community, and it's mission is to provide global collaboration to develop advanced mathematical software for supercomputers.

References

[1] Arnold V.I., Khesin B.A. Topological methods in hydrodynamics. Applied Mathematical Sciences. – Springer-Verlag, N 125, 1998.

[2] Arnold D.N., Falk R.S., Winter R. Finite element exterior calculus, homological techniques, and applications. – Acta Numerica, 2006, 1-155.

[3] Hiptmair R., Li J., Zou J. Universal extension for Sobolev spaces of differential forms and applications. Research Report No. 2009-22, Eidgenossische Technische Hochschule, Zurich, 2009.

[4] Delfour M.C., Zolesio J.P. Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization. – SIAM, Philadelphia, 2011.

[5] Hiptmair R., Li J. Shape derivatives in differential forms: An intrinsic perspective. 2011, Research Report N 2011-42, Eidgenossische Technische Hochschule, CH-8092, Zurich, Switzerland.

[6] Hiptmair R. Finite elements in computational electromagnetism. Acta Numerica, 2002, 237-339.

[7] Hairer E., Lubich Ch., Wanner G. Geometric Numerical Integration. Springer Series in Computational Mathematics. Springer-Verlag, N 31, 2002.

[8] Leimkuhler B., Reich S. Simulating Hamiltonian Dynamics. Number 14 in Cambridge Monographs Series on Applied and Computational Mathematics, Cambridge University Press, 2004.

[9] Il'in V.P. Parallel algorithms for the large applied problems: issues and technologies. //Optoelectronics, instrumentation and data processing. Novosibirsk, vol. 36, N 1, 2000, 3-13.

[10] Chesbrough H. Open Innovation. Harvard Business School Press. -– Boston, Massachusetts, 2003.

[11] Il'in V.P., Skopin I.N. Computational Programming Technologies. Programming and Computer Software. -– Pleides Publishing, Ltd., v. 37, N 4, 2011, 210-222.

[12] Wirth N. Algorithms + Data Structures = Programs. – Prentice-Hall Inc., N. J., 1976.

[13] OpenFOAM. http://www.openfoam.com/.

[14] www.dune-project.org.

[15] Kleppe A. Software Language Engineer. Creating Domain–Specific–Languges Using Metamodels. – Addison–Wesley, Boston, 2008.